

Joint modeling of longitudinal zero-inflated count and time-to-event data: A Bayesian perspective“ by Huirong Zhu, Stacia M. DeSantis and Sheng Luo

OpenBUGS program to fit ZIP joint model in simulation study

The variables created in the program are defined as follows:

n: The total number of observations

N: The number of subjects

pr: The probability of smoking abstinence

pr: The probability of smoking abstinence

mu: The mean of average daily cigarette counts in Poisson distribution

u[i,1]: The random effect related with smoking abstinence for subject i , corresponding to u_i in the joint model

u[i,2]: The random effect related with mean of average daily cigarette counts in Poisson distribution, corresponding to v_i in the joint model

r: The correlation between random effects **u[i,1]** and **u[i,2]**

r1,r2: The coefficients link ZIP and survival submodels, corresponding to ν_1, ν_2 in the joint model

k: The constant baseline hazard in survival submodel,corresponding to h_0 in the manuscript.

The OpenBUGS codes for ZIP joint model are detailed in below.

```

model
{
for (i in 1:n) {
    z[i] <- 0
    z[i] ~ dloglik(l1[i])
    # log-likelihood
    l1[i] <- zero[i]*( log(pr[i] + (1-pr[i])*exp(-mu[i])) + (1-zero[i])
        *(log(1-pr[i]) + y.unb[i]*log(mu[i]) - mu[i] - logfact(y.unb[i]))
    zero[i] <- equals(y.unb[i], 0)
    # two parts for ZIP submodel
    log(mu[i]) <- max(min(beta[1] + beta[2]*trt.unb[i] + beta[3]*week.unb[i] + u[obs[i],1],2),8),-8)
    logit(pr[i]) <- max(min(alpha[1] + alpha[2]*trt.unb[i] + alpha[3]*week.unb[i] + u[obs[i],8]),-8)
}
for (j in 1:N){
    zs[j] <- 0
    phi2[j] <- log(L[j])
    zs[j] ~ dloglik(phi2[j])
    # survival part
    h[j] <- k*exp(gamma*trt[j] + r1*u[j,1] + r2*u[j,2])
    L[j] <- pow(h[j],event[j])*S[j]/1.0E+08
    # event=1 for event; 0 for censored
    S[j] <- exp(-exp(gamma*trt[j] + r1*u[j,1] + r2*u[j,2])*k*time[j])
    u[j, 1:2] ~ dmnorm(zeros[], precision[, ])
}
precision[1:2, 1:2] <- inverse(sigma[, ])
l11 ~ dunif(0.1, 10)
l22 ~ dunif(0.1, 10)
l21 ~ dnorm(0,0.1)
sigma[1, 1] <- pow(l11,2)
sigma[1, 2] <- l11*l21
sigma[2, 1] <- sigma[1, 2]
sigma[2, 2] <- pow(l22,2) + pow(l21,2)
r <- sigma[2,1]/sqrt(sigma[1,1]*sigma[2,2])
sigma1 <- sigma[1,1]
sigma2 <- sigma[2,2]
k ~ dunif(0.01,10)

for (j in 1:nc) {
    beta[j] ~ dnorm(0,0.1)
    alpha[j] ~ dnorm(0,0.1)
}
gamma ~ dnorm(0,0.1)
r1 ~ dnorm(0,0.1)
r2 ~ dnorm(0,0.1)
}

```